Multiscale Analysis of Long Time-series Medical Databases

Shoji Hirano, PhD, and Shusaku Tsumoto, MD, PhD
Department of Medical Informatics, Shimane Medical University, School of Medicine
89–1 Enya-cho, Izumo, Shimane 693–8501, Japan

Abstract

Data mining in time-series medical databases has been receiving considerable attention since it provides a way of revealing useful information hidden in the database; for example relationships between the temporal course of examination results and onset time of diseases. This paper presents a new method for finding similar patterns in temporal sequences based on multiscale matching. Multiscale matching enables us the cross-scale comparison of sequences, namely, it enable us to compare temporal patterns by partially changing observation scales. We examined the usefulness of the method on the chronic hepatitis dataset and found some interesting patterns. On GPT sequences, we found patterns that may represent the effectiveness of interferon (IFN) treatment. On platelet count sequences, we found that, if IFN treatment was ineffective, platelet count kept decreasing following the progress of liver fibrosis, while it started increasing if the treatment was effective.

1 Introduction

Recent advances in medical devices and networking technology enable us to automatically collect huge amount of temporal data on medical laboratory tests, for example blood tests and urinalysis. Analysis of such temporal databases has attracted much interests because it may provide interesting information that can be used to reveal underlying relationships between the temporal patterns of examination results and onset time of diseases. However, despite of its importance, large-scale analysis of time-series medical databases has rarely been performed. This is primarily due to the difficulty in determining appropriate observation scales, i.e., selection of the length of subsequences. Determination of observation scales should be performed carefully, because it directly affects the types of events to be captured. In many cases of practical data mining, this problem is eluded by generating some sets of subsequences changing their lengths, and then performing clustering on each of the sets. However, this approach involves two problems:

(1) *A subsequence may not correctly represent an event.* A subsequence is usually obtained by copying a part of the original sequence that overlaps with a given masking window. The width of the window should be determined in advance, and it should not be changed for all part of the sequences. This means that no feature points of the original sequences, i.e., inflection points and local maxima/minima, are taken into account for determining the shape of the subsequence. Therefore, one cannot guarantee that the subsequence correctly covers the event, namely, whether the head and tail of the subsequence precisely match the start and end of the event respectively.

(2) *Concatenated events of different lengths may not be correctly captured.* Connectivity of subsequences is not guaranteed when the subsequences are obtained using masking windows that have different widths. In other words, there is no guarantee that a set of concatenated subsequences exactly represents a contiguous subpart of the original sequence. This is because no structural information of the original sequence is taken into account in generating the subsequences. Therefore, it is hard to obtain a cluster containing the similar types of concatenated events, i.e., one-week increase followed first by the two-week decrease and then by the one-week increase.

In order to overcome these problems, we propose a grouping method for temporal sequences based on multiscale matching [1]. It compares two temporal sequences by partially changing observation scales. Throughout all scales, it finds the best set of pairs of subsequences under the restrictions that (1) the set contains no miss-matched or over-matched subsequences, and (2) the set minimizes the accumulated differences between the matched subsequences. We also introduce a dissimilarity measure for multiscale comparison of temporal sequences. The dissimilarity measure evaluates dissimilarity of subsequences according to the following aspects: rotation angle (amplitude), length, phase and gradient. We demonstrate the usefulness of this method on the chronic hepatitis datasets. The results show that some interesting temporal patterns showing effectiveness of interferon therapy are discovered.

2 Problems and Related Works

This section describes the problem in time-series medical data analysis and why existing approaches are not suitable for solving the problem.

Usually, a long time-series sequence contains some events that have different durations. Let us consider the case of chronic virus hepatitis as example. The hepatitis C chronically inflames the liver from several years to more than 20 years. In order to evaluate progress of the disease, the temporal course should be observed in long-term observation scales. On the other hand, the anti-viral treatment with interferon is usually applied to the patient during 6 month. Therefore, the change induced by the treatment should be observed in short-term scales. In order to capture both long-term and short-term events, observation scales should be changed partly in the sequence.

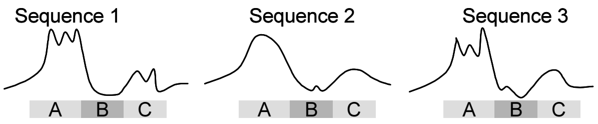

Figure 1: Example sequences.

Figure 1 shows an example of this case. Sequences 1, 2 and 3 are different but they have similar global patterns of increase(A) - decrease(B) - increase(C) when observed in long-term scales. Sequences 1 and 3 have further similar subpatterns in part A. Therefore, when comparing sequences 1 and 3, observation scales at part A should be changed to be shorter than those in sequences 1 and 2 or 2 and 3.

A widely used approach in time-series data mining is to cluster sequences based on the similarity of their primary coefficients. Agrawal et al. [4] utilize discrete Fourier transformation (DFT) coefficients to evaluate similarity of sequences. Chan et al. [5] obtain the similarity based on the frequency components derived by the discrete wavelet transformation (DWT). Korn et al. [6] use singular value decomposition (SVD) to reduce complexity of sequences and compare the sequences according to the similarity of their eigenwaves. Another approach includes comparison of sequences based on the similarity of forms of partial segments. Morinaka et al. [7] propose the L-index, which performs piecewise comparison of linearly approximated subsequences. Keogh et al. [8] propose a method called piecewise aggregate approximation (PAA), which performs fast comparison of subsequences by approximating each subsequence with simple box waves having constant length.

These methods can compare the sequences in various scales of view by choosing proper set of frequency components, or by simply changing size of the window that is used to translate a sequence into a set of simple waves or symbols. However, they are not designed to perform cross-scale comparison. In cross-scale comparison, connectivity of subsequences should be preserved across all levels of discrete scales. Such connectivity is not guaranteed in the existing methods because they do not trace hierarchical structure of partial segments. Therefore, subsequences obtained on the different scales can not be directly merged into the resultant sequences. In other words, one can not capture similarity of sequences by partially changing scales of observation.

3 Multiscale Matching for Time-series Data

Multiscale matching, proposed by Mokhtarian [1], is originally developed as a method for comparing two planar curves by partly changing observation scales. It divides a contour of the object into partial contours based on the place of inflection points. After generating partial contours at various scales for each of the two curves to be compared, it finds the best pairs of partial contours that minimize the total dissimilarity while preserving completeness of the concatenated contours. This method can preserve connectivity of partial contours by tracing hierarchical structure of inflection points on the scale space. Since each ends of a partial contour exactly corresponds to an inflection point and the correspondence between inflection points at different scales are recognized, the connectivity of the partial contours are guaranteed.

We have extended this method so that it can be applied to the comparison of two one-dimensional temporal sequences. A planar curve can be redefined as a temporal sequence, and a partial contour can be analogously redefined as a subsequence. Now let us introduce the basics of multiscale matching for one-dimensional temporal sequence. First, we represent time-series A using multiscale description.

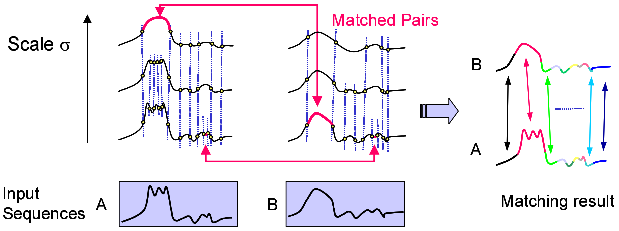

Figure 2: Multiscale matching.

Let $x(t)$ represent an original temporal sequence of A where t denotes a time of data acquisition. The sequence at scale σ, $X(t,\sigma)$, can be represented as a convolution of $x(t)$ and a Gauss function with scale factor σ, $g(t,\sigma)$, as follows:

$$\begin{aligned} X(t,\sigma) &= x(t) \otimes g(t,\sigma) \\ &= \int_{-\infty}^{+\infty} x(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-u)^2/2\sigma^2} du. \quad (1) \end{aligned}$$

Figure 2 shows an example of sequences in various scales. From Figure 2 and the function above, it is obvious that the sequence will be smoothed at higher scale and the number of inflection points is also reduced at higher scale. Curvature of the sequence can be calculated as

$$K(t,\sigma) = \frac{X''}{(1+X'^2)^{3/2}}, \quad (2)$$

where X' and X'' denotes the first- and second-order derivative of $X(t,\sigma)$, respectively. The m-th derivative of $X(t,\sigma)$, $X^{(m)}(t,\sigma)$, is derived as a convolution of $x(t)$ and the m-th order derivative of $g(t,\sigma)$, $g^{(m)}(t,\sigma)$, as

$$X^{(m)}(t,\sigma) = \frac{\partial^m X(t,\sigma)}{\partial t^m} = x(t) \otimes g^{(m)}(t,\sigma). \quad (3)$$

The next step is to find inflection points according to change of the sign of the curvature and to construct segments. A segment is a subsequence whose ends respectively correspond to the adjacent inflection points. Let $\mathbf{A}^{(k)}$ be a set of N segments that represents the sequence at scale $\sigma^{(k)}$. $\mathbf{A}^{(k)}$ can be represented as

$$\mathbf{A}^{(k)} = \left\{ a_i^{(k)} \mid i = 1, 2, \cdots, N^{(k)} \right\}. \quad (4)$$

In the same way, for another temporal sequence B, we can obtain a set of segments $\mathbf{B}^{(h)}$ at scale $\sigma^{(h)}$ as

$$\mathbf{B}^{(h)} = \left\{ b_j^{(h)} \mid j = 1, 2, \cdots, M^{(h)} \right\}, \quad (5)$$

where M denotes the number of segments of B at scale $\sigma^{(h)}$.

The main procedure of multiscale structure matching is to find the best set of segment pairs that minimizes the total difference. Figure 2 illustrates the process. For example, five contiguous segments at the lowest scale of Sequence A

are integrated into one segment at the highest scale, and the integrated segments well match to one segment in Sequence B at the lowest scale. Thus the set of the five segments in Sequence A and the one segment in Sequence B will be considered as a candidate for corresponding subsequences. While, another pair of segments will be matched at the lowest scale. In this way, matching is performed throughout all scales. The resultant set of segment pairs must not be redundant or insufficient to represent the original sequences. Namely, by concatenating all subsequences in the set, the original sequence must be completely reconstructed without any partial intervals or overlaps. The matching process can be fasten by implementing dynamic programming scheme [2].

The total difference between sequences A and B is defied as a sum of dissimilarities of all matched segment pairs as

$$D(A,B) = \sum_{p=1}^{P} d(a_p^{(0)}, b_p^{(0)}), \tag{6}$$

where P denotes the number of matched segment pairs. The notation $d(a_i^{(k)}, b_j^{(h)})$ denotes dissimilarity of segment pairs $a_i^{(k)}$ and $b_j^{(h)}$ at scales k and h defined below.

$$d(a_i^{(k)}, b_j^{(h)}) = \max(\theta, l, \phi, g), \tag{7}$$

where θ, l, ϕ and g respectively denote segment difference on rotation angle, length, phase and gradient defined below.

$$\theta(a_i^{(k)}, b_j^{(h)}) = \frac{\mid \theta_{a_i}^{(k)} - \theta_{b_j}^{(h)} \mid}{\theta_{a_i}^{(k)} + \theta_{b_j}^{(h)}} \tag{8}$$

$$l(a_i^{(k)}, b_j^{(h)}) = \left| \frac{l_{a_i}^{(k)}}{L_A^{(k)}} - \frac{l_{b_j}^{(h)}}{L_B^{(h)}} \right|, \tag{9}$$

$$\phi(a_i^{(k)}, b_j^{(h)}) = \left| \frac{\phi_{a_i}^{(k)}}{\Phi_A^{(k)}} - \frac{\phi_{b_j}^{(h)}}{\Phi_B^{(h)}} \right|, \tag{10}$$

$$g(a_i^{(k)}, b_j^{(h)}) = \begin{cases} 1, & \text{if } g_{a_i}^{(k)} \times g_{b_j}^{(h)} < 0 \\ \left| g_{a_i}^{(k)} - g_{b_j}^{(h)} \right|, & \text{otherwise.} \end{cases} \tag{11}$$

where $\theta_{a_i}^{(k)}$ and $\theta_{b_j}^{(h)}$ denote rotation angles of tangent vectors along the contours, $l_{a_i}^{(k)}$ and $l_{b_j}^{(h)}$ denote length of the contours, $L_A^{(k)}$ and $L_B^{(h)}$ denote total segment length of the sequences A and B at scales $\sigma^{(k)}$ and $\sigma^{(h)}$, $\phi_{a_i}^{(k)}$ denotes temporal delay from the first time of data acquisition, $\Phi_A^{(k)}$ denotes durations of data acquisition of sequence A, $g_{a_i}^{(k)}$ denotes difference of data values at both ends of segment $a_i^{(k)}$, and σ denotes standard deviation of the data values.

The first two terms θ and l defined in Equations (8) and (9) characterize shapes of the subsequences. Large differences can be assigned when difference of rotation angle (amplitude) or relative length is large. The third term ϕ defined in Equation (10) emphasizes difference on phase. Phase $\phi_{a_i}^{(k)}$ is defined as an acquisition time t of the head point of segment $a_i^{(k)}$. It will be normalized by the acquisition durations $\Phi_A^{(k)}$ before taking subtraction to $\phi_{b_j}^{(h)}/\Phi_B^{(h)}$. The last term g defined in Equation (11) emphasizes difference on gradient normalized by the standard deviation of the corresponding attribute value. Figure 3 illustrates meaning of these terms.

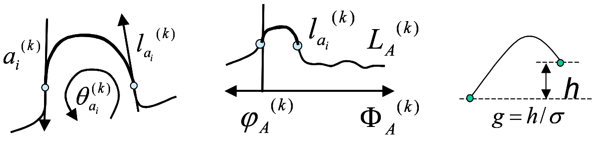

Figure 3: Components of the dissimilarity measure.

By this measure, we simultaneously evaluate dissimilarity of two events from the following aspects: (1) intenseness of increase/decrease of data values, (2) length of the events (3) dates of the events (4) global trends of the events. Besides, by taking maximum of these four factors, we improve discrepancy between sequences.

4 Experimental Results

We examined usefulness of this method on the chronic hepatitis dataset [9], which was used as a common dataset at ECML/PKDD Discovery Challenge 2002. The dataset contained long time-series data on laboratory examinations, which were collected at Chiba University Hospital in Japan. The subjects were 771 patients of hepatitis B and C who took examinations between 1982 and 2001. Each sequence originally had different sampling intervals from one day to one year. From preliminary analysis we found that the most frequently appeared interval was one week; this means that most of the patients took examinations on a fixed day of a week. According to this observation, we determined resampling interval to seven days.

First, we applied the proposed method to the GPT sequences. Here we removed 268 of 771 sequences because biopsy information was not provided for them and thus their virus types were not clearly specified. We performed multiscale matching for every pair of sequences and then performed rough clustering [3] of the sequences using the derived dissimilarities. The resultant clusters were stratified according to the virus type and administration of the interferon (IFN) treatment. For clusters that had interesting compositions, we visually inspected common patterns in those clusters.

Clusters well reflected effectiveness of the interferon treatment. Table 4 shows a part of the clustering result. Two types of interesting patterns were found in the clustered sequences. The first pattern was found in cluster 4, which contained remarkably many cases of type C with IFN (B/C/C(IFN) = 6/3/25). In this cluster, GPT decreased after administration of IFN and then kept flattened at low level (figure 4). This pattern represented cases where interferon successfully suppressed activity of the type C hepatitis virus. The second pattern was found in clusters 1, 5 and 7. In these clusters, GPT had continuous vibrations (figure 5). Since this pattern was commonly observed regardless of virus type and administration of IFN, it implied ineffec-

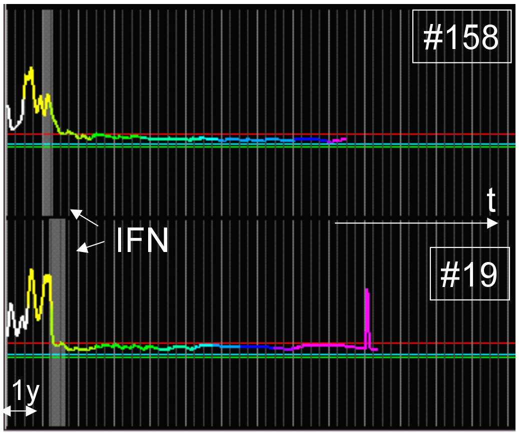

Figure 4: GPT cluster 4: #19(type C; IFN), #158(type C; IFN)

tive cases of IFN treatment. Note that figures 4 and 5 represent matching results of the sequences grouped into the same cluster. The sequence number is shown as #xxx and the matched subsequences are painted in the same color.

Table 1: Clusters of GPT sequences

cluster	IFN=N		IFN=Y	total
	B	C	C	
1	24	13	42	79
2	9		7	16
3	44	25	24	93
4	6	3	25	34
5	5	4	6	15
6	1		2	3
7	42	19	31	92
⋮				
44		1		1
total	206	100	197	503

Next, we applied the proposed method to the sequences of platelet count (PLT) in the hepatitis data set. The dataset contained 722 sequences but 219 of them had no information about virus type. We excluded them and examined the remaining 503 PLT sequences. The procedure of analysis was the same as GPT sequences.

We found interesting clusters of PLT sequences on patients of type C hepatitis who had been applied IFN treatments. These clusters contained sequences that took common chronic courses as shown in figures 6 and 7. From sequences in figure 6 it could be seen that PLT increased after completion of IFN treatment. This pattern might represent a typical case where ability of producing platelet had been recovered as the liver had been cured by the IFN treatment. On the contrary, sequences in figure 7 showed a pattern in which PLT chronically kept decreasing even after completion of IFN treatment. These two types of patterns suggested that PLT increased when IFN treatment was effective, and PLT kept decreasing when IFN treatment was ineffective and it resulted in bleeding.

For clusters that contained ineffective cases of IFN treatment, we further examined relationships between stage of liver fibrosis and years until platelet count recedes from its normal range. The results, 0-15 years for F1, 0-10 years for F2, 0-8 years for F3 and F4, had correspondence to those of natural courses of the chronic hepatitis to which IFN treatment was not applied. These results implied that patients who received ineffective IFN treatment and who did not receive IFN treatment represent similar chronic courses on platelet counts.

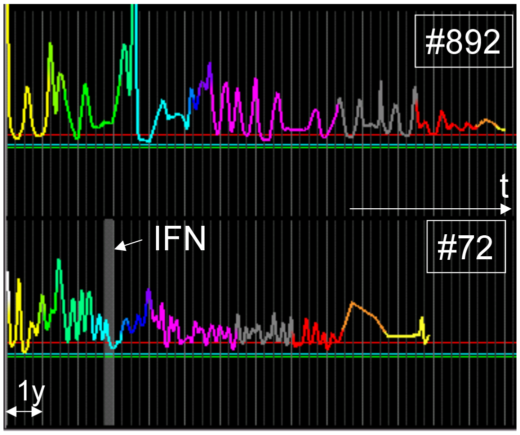

Figure 5: GPT cluster 1: #72(type C; IFN), #892(type B)

Table 2: Relationship between stage of liver fibrosis and platelet count

Virus Type	N	Stage of Fibrosis	Av. PLT	SD PLT
B	61	F1	206.76	51.79
B	51	F2	173.45	44.40
B	25	F3	163.84	45.91
B	22	F4	114.73	50.10
C	21	F0	232.76	63.48
C	38	F1	186.83	54.19
C	81	F2	150.85	47.58
C	67	F3	137.45	44.41
C	62	F4	123.76	45.00

Based on these observations, we extended subjects to all patients of type B and type C virus hepatitis, and examined relationships between stage of liver fibrosis and platelet count using original data. Table 2 shows the results. For both types, it can be seen that progress of liver fibrosis had high correspondence to decrease of platelets.

Table 3 shows relations between activity of virus, stage of liver fibrosis and years until platelet count recedes from its normal range. It can be seen that platelet counts rapidly become abnormal in the patient who had higher stage of fibrosis. If their stages are the same, the ones who had high virus activities receded from normal range faster.

5 Conclusions

In this paper, we have presented a new analysis method of long time-series medical databases based on the multiscale matching. The method enabled us to compare sequences by partly changing observation scales. Experi-

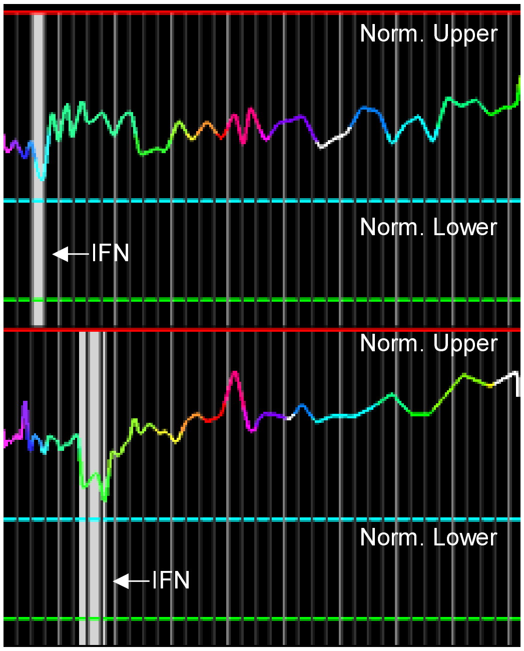

Figure 6: Matching result of PLT sequences that have V-shape trends

Table 3: Relations between stage of liver fibrosis, virus activity and time until platelet count becomes abnormal

Fibrosis	Activity	N	Time	SD Time
F1	A1	9	5.24	4.32
F1	A2	6	3.36	3.95
F2	A1	1	–	–
F2	A2	16	2.87	3.25
F3	A1	1	–	–
F3	A2	5	3.67	3.59
F3	A3	9	0.68	1.02
F4	A1	1	–	–
F4	A2	15	0.88	2.27
F4	A3	9	0.08	0.17

ments on the chronic hepatitis data showed the usefulness of this method. For GPT sequences, we found interesting patterns that may represent effectiveness of IFN treatment. For PLT sequences, we found that, if IFN treatment was ineffective, platelet count kept decreasing following the progress of liver fibrosis, while it started increasing if the treatment was effective. These results suggested the possibility of using blood tests for predicting stage of liver fibrosis and effectiveness of IFN treatment as an alternative of invasive liver biopsy. In the future we would like to clinically validate these hypotheses.

Acknowledgment

This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Area (B)(No.759) "Implementation of Active Mining in the Era of Information Flood" by the Ministry of Education, Culture, Science and Technology of Japan.

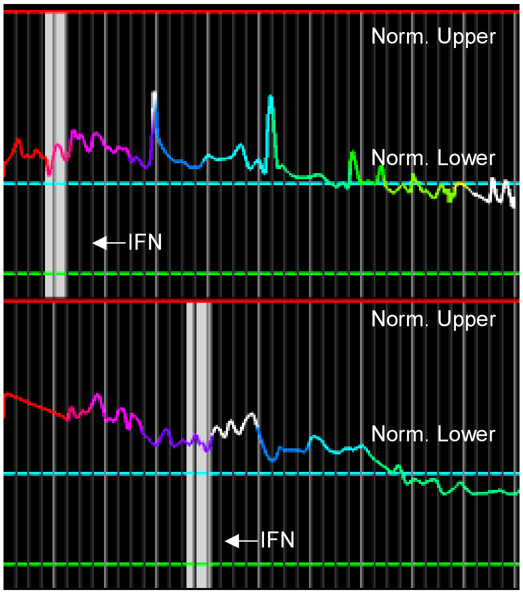

Figure 7: Matching result of PLT sequences that have decreasing trends

References

[1] F. Mokhtarian and A. K. Mackworth (1986): Scale-based Description and Recognition of planar Curves and Two Dimensional Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(1): 24-43.

[2] N. Ueda and S. Suzuki (1990): A Matching Algorithm of Deformed Planar Curves Using Multiscale Convex/Concave Structures. IEICE Transactions on Information and Systems, J73-D-II(7): 992–1000.

[3] S. Hirano and S. Tsumoto (2003): An Indiscernibility-based Clustering Method with Iterative Refinement of Equivalence Relations. Journal of Advanced Computational Intelligence and Intelligent Informatics (to appear).

[4] R Agrawal, C. Faloutsos, and A. N. Swami (1993): Efficient Similarity Search in Sequence Databases. Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms: 69–84.

[5] K. P. Chan and A. W. Fu (1999): Efficient Time Series Matching by Wavelets. Proceedings of the 15th IEEE International Conference on Data Engineering: 126–133.

[6] F. Korn, H. V. Jagadish, and C. Faloutsos (1997): Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. Proceedings of ACM SIGMOD International Conference on Management of Data: 289–300.

[7] Y. Morinaka, M. Yoshikawa, T. Amagasa and S.Uemura (2001): The L-index: An Indexing Structure for Efficient Subsequence Matching in Time Sequence Databases. Proceedings of International Workshop on Mining Spatial and Temporal Data, PAKDD-2001: 51-60.

[8] E. J. Keogh, K. Chakrabarti, M. J. Pazzani, and S. Mehrotra (2001): "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases" Knowledge and Information Systems 3(3): 263-286.

[9] URL: http://lisp.vse.cz/challenge/ecmlpkdd2002/